Development of a hand pose recognition system on an embedded computer using Artificial Intelligence

Dennis Núñez Fernández
Universidad Nacional de Ingeniería
Lima, Peru
dnunezf@uni.pe

***Abstract*—The recognition of hand gestures is a very interesting research topic due to the growing demand in recent years in robotics, virtual reality, autonomous driving systems, human-machine interfaces and in other new technologies. Despite several approaches for a robust recognition system, gesture recognition based on visual perception has many advantages over devices such as sensors, or electronic gloves. This paper describes the implementation of a visual-based recognition system on a embedded computer for 10 hand poses recognition. Hand detection is achieved using a tracking algorithm and classification by a light convolutional neural network. Results show an accuracy of 94.50%, a low power consumption and a near real-time response. Thereby, the proposed system could be applied in a large range of applications, from robotics to entertainment.**

***Index Terms*—Gesture Recognition, Human-Machine Interaction, Recognition System, Hand Poses, Embedded Computer.**

I. Introduction

Hand gesture recognition is one obvious strategy to build user-friendly interfaces between machines and users. In the near future, hand posture recognition technology would allow for the operation of complex machines and smart devices through only series of hand postures, finger and hand movements, eliminating the need for physical contact between man and machine. Hand gesture recognition on images from common single camera is a difficult problem because occlusions, variations of posture appearance, differences in hand anatomy, etc. Despite these difficulties, several approaches to gesture recognition on color images has been proposed during the last decade [1].

In recent years, Convolutional Neural Networks (CNNs) have become the state-of-the-art for object recognition in computer vision [2]. In spite of high potential of CNNs in object detection problems [3] [4] and image segmentation [2] tasks, only few papers report successful results (a recent survey on hand gesture recognition [1] reports only one important work [5]). Some obstacles to wider use of CNNs are high computational costs, lack of sufficiently large datasets, as well as lack of hand detectors appropriate for CNN-based classifiers. In [6], a CNN has been used for classification of six hand poses to control robots using colored gloves. In more recent work [7], a CNN has been implemented on the Nao robot. In a recent work [8], a CNN has been trained on one million of images. However, only a portion of the dataset with 3361 manually labeled frames in 45 classes of sign language is publicly available.

In this work we developed a system for hand pose recognition to work on embedded computers with limited computational resources and making use of low power consumption. In order to accomplish these targets, we employ low-processing algorithms and trained a light CNN, which was optimized to balance high accuracy, fast time response, low power consumption and low computational costs.

II. Methodology

A. Overview

The proposed system works with images captured from a standard CMOS camera and executed on embedded computers with low computational resources, without GPU support, such as Raspberry Pi, BeagleBone Board, Banana Pi, Intel Galileo Board, and others. Therefore, the main objectives of the proposed system are as follows: high accuracy rate, fast time response, low power consumption and low computational costs.

The system is composed of three main steps: hand detection, hand region tracking and hand gesture recognition. In the first step the Haar cascades classifier detects a basic hand shape in order to have a good hand detection. Then, this hand region is tracked using the MIL (Multiple Instance Learning) tracking algorithm. Finally, hand gesture recognition is performed based on a trained Convolutional Neural Network. Since the steps described before are designed to consume few computational resources, the whole system will be implemented on a personal computer and Raspberry Pi board. Fig. 1 shows the steps mentioned above.

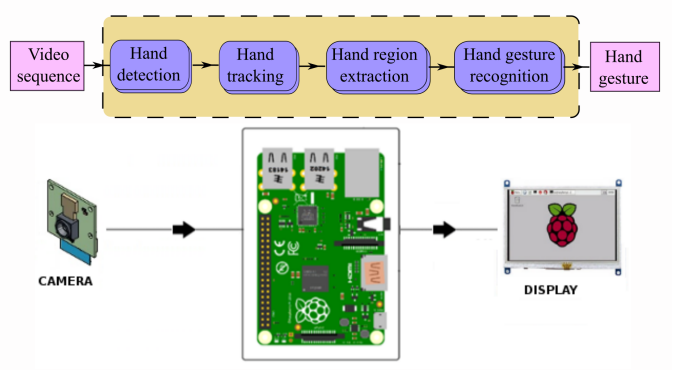


Fig. 1: Diagram for the proposed system

978-1-7281-3646-2/19/$31.00 ©2019 IEEE

B. Haar Cascades Classifier

The Viola-Jones object detection algorithm is the first object detection algorithm to provide competitive object detection rates in real-time. Although it can be trained to detect a variety of object classes, it was mainly intended by the problem of hand detection.

This approach to hand detection combines four key concepts: Haar-like features: simple rectangular features, Integral image: for rapid feature detection, AdaBoost: machine-learning method, Cascade classifier: to combine many features efficiently, see Fig. 2.

Fig. 2: Haar cascade classifier

C. Hand Tracking

Haar cascade classifier allows better detection of objects with static features such as balloons, boxes, faces, eyes, mounts, noise, etc. But a hand in motion has few static features because its shape and fingers can change as well as its orientations. So, Haar cascade classifier allows detection of only basic hand poses, which are not suitable to recognize a hand in motion with a long mount of different poses.

Since hand detection using Haar cascades in not a robust method, this deficiency is compensated with a hand tracker based on wrist region. Furthermore, wrist region is proposed for tracking due this region keeps invariant and has static features when hand changes to different poses, shapes and orientations.

In addition to this, hand tracking allows the reduction of the processing time since tracking requires less computational resources than hand detection (whole image evaluation versus local evaluation). Fig. 3 shows the different hand regions used for detection and tracking, as image shows the hand region for tracking (blue box) encloses the hand in different shapes and poses. Therefore, the hand region inside the blue box will be used by the CNN to perform the hand gesture recognition.

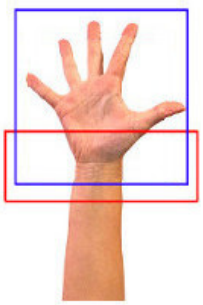


Fig. 3: Wrist region for detection (red box) and hand region for tracking (blue box)

In this project, the MIL (Multiple Instance Learning) algorithm will be used for hand tracking. The MIL algorithm trains a classifier in an online manner to separate the object from the background. Multiple Instance Learning avoids the drift problem for a robust tracking. So, MIL results in a more robust and stable tracker. The implementation is based on [9]. In addition to this, the proposed tracking algorithm consumes less memory and computational resources than the Haar cascade classifier.

D. Skin Detection

Skin color is a powerful feature for fast hand detection. Essentially, all skin color-based methodologies try to learn a skin color distribution, and then use it to extract the hand region. In this work the hand region has been obtained on the basis of statistical color models [10]. A model in RGB-H-CbCr color spaces has been constructed on the basis of a training dataset. Later, the hand probability image has been thresholded. Finally, after morphological closing, a connected components labeling has been executed to extract the gravity center of the region, coordinates of the most top pixel as well as coordinates of the most left pixel of the hand region.

E. Hand Poses Dataset

The dataset for hand gesture classification was provided by an open database from AGH University of Science and Technology [11]. This is composed of 73,124 grayscale images of size 48x48 pixels divided into ten different hand gestures, captured from ten persons of different nationalities. However, for purposes of this project, only binary images were selected. From this dataset, the 80% (42,027 images) were used for training and 20% (14,667 images) for testing. Fig. 4 depicts samples of each hand gesture, also called class. The principal advantage of this dataset is that the hands were approximately aligned in such a way that characteristic hand features (e.g. the wrist) are approximately located at pre-defined positions in the image. This means that in each class the wrists are approximately located at the same position. Furthermore, thanks to such an method the recognition of hand poses at acceptable frame rates can be succeeded with a simple convolutional neural network and at a lower computational cost.

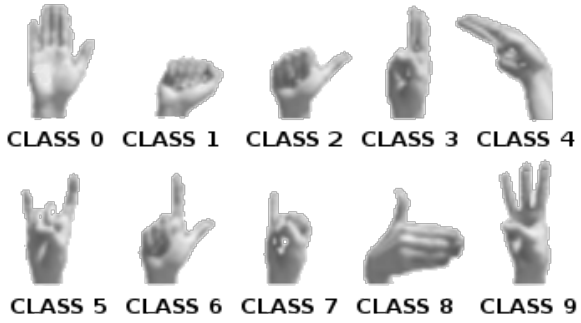


Fig. 4: Sample images of hand gestures used in training and testing steps

F. Convolutional Neural Networks

Since each hand pose is composed of strongly different strokes, recognition doesnt need large images and complex CNNs to extract useful features. In this way, we only use binary images of 48x48 pixels, and a small CNN with few layers and parameters.

The proposed CNN is formed by two convolutional layers with kernels of 5x5 and 3x3 size each one, a non linearity (ReLU) activation function and a max-pooling layer after every convolutional layer, and two full-connected (FC) layers of 120 neurons length followed by a final 10-way softmax, see Fig. 5. Furthermore, this CNN is composed by only 60K learnable parameters. This number of parameters are significantly less than the AlexNet network (60M learnable parameters and 650,000 neurons) [2] and the GoogleNet (6.8M learnable parameters) [12].

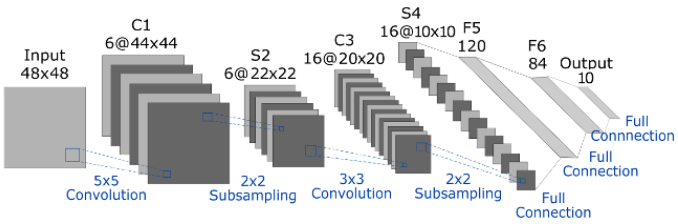


Fig. 5: Architecture of the CNN for hand pose recognition.

III. Experimental Results

A. Experimental Results of the Model

The performance of the CNN of hand poses classification was evaluated using different metrics such as confusion matrix and accuracy. The confusion matrix presents a visualization of the misclassified classes and helps to add more training images in order to improve the model. The confusion matrix of our model is shown in Fig. 6 and discloses which letters are misclassified. These errors happen because of similarities between the classes. Furthermore, our architecture shows an outstanding accuracy of 94.50%.

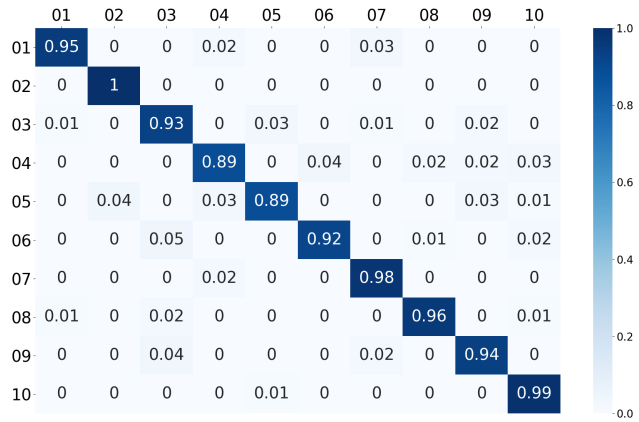
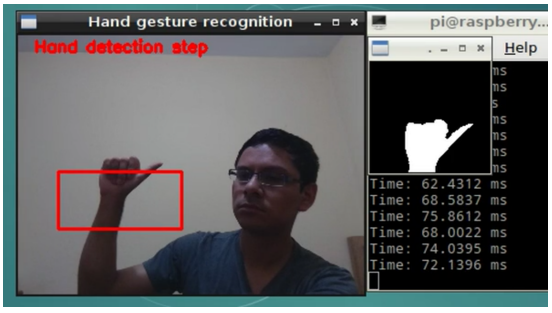


Fig. 6: Confusion matrix

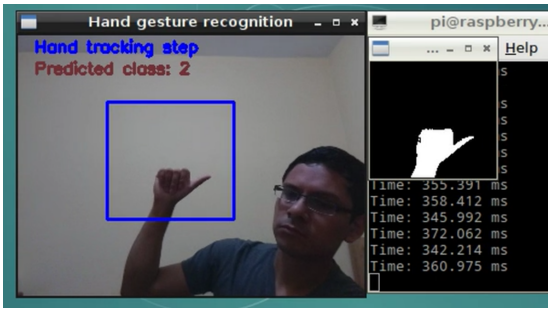
B. Experimental Results of Inference

The implementation of the proposed recognition system on a personal computer has no issues due to its high computational resources. However, when a recognition system is implemented on embedded computers like the Raspberry Pi 3 we have two major obstacles working against us: limited RAM memory (only 1 GB) and restricted processor speed (four ARM Cortex-A53 @1. 2 GHz). In order to obtain a better computation performance, the system was implemented using C++ language.

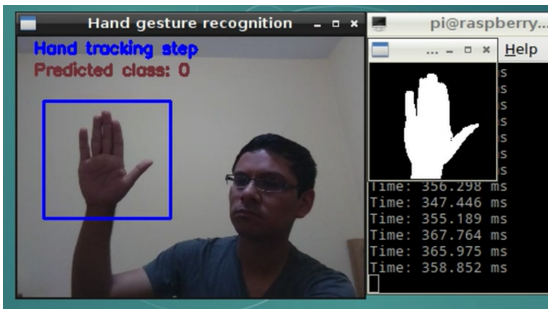
In spite of the processing and memory limitations mentioned above, our real-time recognition system shows promising results during the evaluation step. Fig. 7 depicts its performance in real environments on images of 640x480 pixels. As you can observe, the system correctly recognizes different hand poses, although some shape distortions, low light conditions, and different sizes. In addition to this, we obtain a fast response time of about 351.2 ms (average of 100 iterations) to detect and classify a single hand pose.



(a) Hand detection



(b) Hand tracking and pose recognition



(c) Hand tracking and pose recognition

Fig. 7: Results of hand pose recognition on a Raspberry Pi 3

The Table I shows some details of CNNs tested on the Raspberry Pi 3 platform. As you can see, the proposed CNN achieves the fastest time response, compared with other architectures, by using the lowest power consumption because its simple and efficient design.

TABLE I: Response time and power consumption for evaluation of different CNNs on a Raspberry Pi 3 using Caffe

Model	Proposed CNN	VGG_F [13]	NiN [14]	AlexNet [2]	GoogLeNet [12]
Layers	9	13	16	11	27
Power (W.)	0.690	0.760	0.840	0.750	0.790
Time (s.)	0.351	0.857	0.553	1.803	1.175

IV. Conclusion

In this paper, we introduce the implementation of a hand pose recognition system on a regular embedded computer. We demonstrated that our system is capable to recognize 10 hand gestures with an accuracy of 94.50% on images captured from a single RGB camera, and using low power consumption, about 0.690 W. In addition, the average time to process each 640x480 image on a Raspberry Pi 3 board is 351.2 ms. The results demonstrate that our recognition system is suitable for embedded applications in robotics, virtual reality, autonomous driving systems, human-machine interfaces and others.

References

[1] Oyedotun, O., Khashman, A.: Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications (2016) 111

[2] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS. (2012) 10971105

[3] Kwolek, B.: Face detection using convolutional neural networks and Gabor filters. In: Int. Conf. Artificial Neural Networks, LNCS, vol. 3696, Springer (2005) 551556

[4] Arel, I., Rose, D., Karnowski, T.: Research frontier: Deep machine learninga new frontier in artificial intelligence research. Comp. Intell. Mag. 5(4) (2010) 1318

[5] Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans. Graph. 33(5) (2014)

[6] Nagi, J., Ducatelle, F.: Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: IEEE ICSIP. (2011) 342347

[7] Barros, P., Magg, S., Weber, C., Wermter, S.: A multichannel convolutional neural network for hand posture recognition. In: 24th Int. Conf. on Artificial Neural Networks (ICANN), Cham, Springer (2014) 403410

[8] Koller, O., Ney, H., Bowden, R.: Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In: IEEE Conf. on Comp. Vision and Pattern Rec. (2016) 37933802

[9] B Babenko, M-H Yang, S Belongie, Visual Tracking with Online Multiple Instance Learning, IEEE CVPR09, June, 2009.

[10] Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. Int. J. Comput. Vision 46(1) (2002) 8196

[11] Núñez Fernández D., Kwolek B., Hand Posture Recognition Using Convolutional Neural Network. In: Mendoza M., Velastn S. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2017. Lecture Notes in Computer Science, vol 10657. Springer, Cham

[12] C. Szegedy et al., Going deeper with convolutions, 2015 IEEE Conf. on Computer Vision and Pattern Recogn. (CVPR), Boston, MA, 2015.

[13] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In Proceedings of the British Machine Vision Conference 2014, pages 6.16.12. British Machine Vision Association, 2014.

[14] Min Lin, Qiang Chen, and Shuicheng Yan. Network In Network. In International Conference on Learning Representations (ICLR) 2014.